

**Amendments to the Specification:**

Please amend specification to conform to the amendment to the drawings as follows:

Page 2, replace the entire paragraph starting at line 29 with the following new paragraph:

-- ~~Figs. 2a and 2b~~ Fig. 2 illustrates examples of portions of the software function of the system of Fig. 1; and--

Page 3, replace the entire paragraph starting at line 1 with the following new paragraphs

~~Figs. 3-6 are software flow diagrams and sequence diagram illustrating two examples of the operation of the system of Fig. 1.~~ Figs. 3A-3B is a software flow diagram illustrating the process of getting translated names and values of all variables;

Figs. 4A-4B is a software flow diagram illustrating the process of changing the value of a variable;

Figs. 5A-5D is a software sequence diagram corresponding to the flow diagram of Figs. 3A-3B; and

Figs. 6A-6D is a software sequence diagram corresponding to the flow diagram of Figs. 4A-4D. --

Page 5, replace the entire paragraph starting at line 13 with the following new paragraph:

Fig. 2[[a]] is an example of formatted data from the data dictionary 24 or data central 26 with variable descriptions in XML based language. The <VARIABLES> may contain multiple <VARIABLE> elements. Each <VARIABLE> element has a "KEY" attribute and multiple child <NAME> elements with translations. Fig. 2[[b]] also shows [[is]] an example of formatted data from a data agent 20 formatted in XML. The root XML element is <VARIABLES> which may contain multiple <VARIABLE> elements with "NAME", "VALUE" or "OPTIONS" attributes and with <OPTION> child elements with various possible variable values. Both of the examples of Fig. 2a ~~and Fig. 2b~~ are provided as samples. Real syntax may vary depending upon the requirements of a particular application.--

Page 5, replace the entire paragraph starting at line 22 with the following:

-- Figs. 3A-3B and 4A-4B are standard program flow diagrams using module: function descriptions in the oval boxes with the diamond shaped boxes being branching

points. In Figs. 3A-3B and 4A-4B, the module names are abbreviated as follows:

UI     - User Interface Module  
DD     - Data Dictionary Module  
DA     - A Data Agent Module  
DE     - The Data Engine Module  
DC     - Data Central --

Page 5, replace the entire paragraph starting at line 30 with the following new paragraph:

-- Figs. 5A-5D and 6A-6D are sequence diagrams in standard Unified Modeling Language (UML). In Figs. 5A-5D and 6A-6D, the horizontal boxes across the top and the module names as used above. In Figs. 5A-5D and 6A-6D time is shown as progressing from the top of the diagrams to the bottom and the arrows represent messages passing among the modules. The vertical boxes or dashed vertical lines represent the time of the module activity.

Page 6, replace the entire paragraph starting at line 5 with the following new paragraph:

-- Figs. 3A-3B and 5A-5D, taken together are an example of the process of getting translated names and values of all variables. At step 302, the user interface requires values for all variables for each device at a given address using a selective language. At step 304, the data engines selects the particular data agent based upon the address for the device. At step 306, the data dictionary provides the identification names of variables based upon the data agent and address. If the required data is not in the data dictionary, the data central gets the data at block 308 and updates the data dictionary at block 310. At block 312, the data agent provide values from the required address for selected variable names. If the data agent can not provide the required data, the data engine tries another data agent at block 304 and the process repeats. At block 314, the data engine determines the device type from the values of the variables and at block 316, the data dictionary provides names of all of the variables that are relevant for the particular type of device and for the particular data agent. If the data dictionary does not have the list of variables, the names are obtained from the data central at block 318 and, at block 320, the data dictionary is updated. At block 322, the data agent gets the values of the variables

for the list of all variables. At block 324, the data dictionary translates the values to the selected language. If insufficient information is available in the data dictionary to perform the translation, the information is obtained from data central at block 326 and at block 328, the data dictionary is updated with the newly obtained data. Finally, at block 330, the user interface module is presented with the translated names and values of all of the variables. The same sequence of steps can be followed by referring to the UML sequence diagram of Figs. 5A-5D. --

Page 6, replace the entire paragraph starting at line 25 with the following new paragraph:

-- Figs. 4A-4B and 6A-6D illustrate the steps involved in changing a value of a variable. At block 402, the user interface module selects a device address and variable key. At block 404, the data engine selects the agent based upon the selected address. If the agent is available, at block 406, the data dictionary provides the variable name for a given variable key and agent and at block 408 the data agent provides possible values for the variable. If there is no available data agent, at block 410 the data engine tries to determine the device type based upon the address and attempts to get possible values at block 412. If the data dictionary does not contain the possible values they are obtained from the data central at block 414 and the data dictionary is updated at block 416. At block 418, the data dictionary translates the possible variable values to the selected language. If the data dictionary does not have the information necessary for the translation, at block 420 the information is obtained from data central and at block 422, the data dictionary is updated. At block 424, the user interface presents translated possible values of the variables and the user selects the new value. At block 426, the data engine again selects the agent and at block 428, the data dictionary translate the new variable value to the selected data agent language. If the data dictionary does not have sufficient information to complete the translation, at block 430, the translation information is obtained from data central and at block 432, the data dictionary is updated with the new information. At block 434, the data agent changes the variable value. The process describe above and as shown in connection with Figs. 4A-4B can be followed by reference to the UML sequence diagram of Figs. 6A-6D. --